INDEPENDENT DESKEW LANE

NOTICE OF COPYRIGHTS AND TRADE DRESS

BACKGROUND

Field Of The Invention

[0002] The present invention relates to high speed data communications.

Related Art

[0003] High speed data communications over optical links is governed by various standards. One of the standards is the *Serdes Framer Interface Level 5 (SFI-5): Implementation Agreement for 40Gb/s Interface for Physical Layer Devices* promulgated by the Optical Internetworking Forum of Fremont, California. Serdes refers to SERialization and DESerialization. This standard, referred to herein as the SFI-5 standard, specifies the physical layer requirement for communications at 40 gigabits per second (Gb/s). The SFI-5 standard defines the interface between a Serdes component, a forward error correction unit, and a framer. The SFI-5 standard specifies 16 bit wide data bus, that is, a data bus with 16 data lanes. A deskew lane is also specified as a seventeenth lane.

[0004] The data signals may encounter different delays in transit from the SFI-5 source device to the sink device. The earliest arriving signal may lead the latest arriving by n bits. Relative to the earliest, each of the remaining signals is coincident, or is up to n unit intervals late. The search space for determining the relative delays of all 17 signals on SFI-5 is $(n + 1)^{17}$ combinations. The deskew lane serves as a reference lane to allow each of the 16 data lanes to independently measure its own delay relative to the reference signal.

[0005] The deskew function is shared between the SFI-5 source and sink devices at either end of receive and transmit interfaces. In a source device, data is sampled from each of the 16 data lanes sequentially, and copied onto the deskew lane. The deskew lane is then sent with the 16 data lanes to the sink device over the SFI-5 interface. Data input to the sink device may be skewed by the different delays in each of the data lanes. It is the function of a deskew algorithm operating in the sink device to measure the amount of skew on each data channel, and then to use this skew information to compensate for the amount of skew. The deskew algorithm may make an initial skew measurement on initial power up or connection. Subsequent to skew measurement, external conditions may vary causing the skews to change. SFI-5 compliant devices track skew changes without introducing data or transmission errors. The deskew algorithm may operate continuously during normal operation of an SFI-5 interface to continuously track skew.

[0006] Testing systems may be used to verify whether systems, devices and apparatuses conform to the SFI-5 standard and to evaluate whether SFI-5 compliant systems, devices and apparatuses can handle anticipated or large loads of network traffic.

THE DRAWINGS

[0007] FIG. 1 is a block diagram of an environment in which an independent deskew lane may be practiced.

[0008] FIG. 2 is a block diagram of a lane card.

[0009] FIG. 3 is a block diagram of transmitter circuit in which an independent deskew lane may be practiced.

[0010] FIG. 4 is a block diagram of receiver circuit in which an independent deskew lane may be practiced.

[0011] FIG. 5 is a flow chart of the actions taken when transmitting data using an independent deskew lane described herein.

[0012] FIG. 6 is a flow chart of the actions taken when receiving data according using an independent deskew lane described herein.

DETAILED DESCRIPTION

[0013] Throughout this description, the embodiments and examples shown should be considered as exemplars, rather than limitations on the apparatus and methods of the present invention.

A System

[0014] FIG. 1 is a block diagram of an environment in which an independent deskew lane may be practiced. A testing system 110 may be coupled with a device under test (DUT) or a system under test (SUT). The DUT/SUT may be an optical communications unit, a communications equipment component, including boards or cards having computer chips, and communications chips, or a combination of these, all of may support a communications standard, and, more specifically, an optical communications standard. As shown, DUT 140 is a computer chip which is located on a test board 142. The DUT 140 may be coupled to the testing system 110 via connector 130. The DUT 140 may provide support for and conform to the SFI-5 standard, other physical layer optical communications standard, or other physical (PHY) layer communications standard. The test board 142 may include an optical fiber loop back 144. As shown, the DUT 140 may be a computer chip such as, for example, a field programmable gate array (FPGA), an application specific integrated circuit (ASIC), an electrically erasable programmable memory (EEPROM) chip, a programmable logic array (PLA), a programmable logic device (PLD), and others.

[0015] The testing system 110 may be used to test whether the DUT 140 or other DUT or SUT conforms to a PHY layer optical communications standard such as SFI-5 or a PHY layer

communications standard. The testing system 110 may be a bit error rate testing (BERT) system. The testing system 110 may provide for 16 data lanes 126 and one deskew lane 128 as required by the SFI-5 specification. The data and deskew lanes may be included in a single cable and as 17 separate cables. The testing system 110 may operate to test DUTs at 10Gbps, 40 Gbps, and slower and faster. In one embodiment, each data lane 126 operates at 2.5 Gbps. Other data lane speeds may be used to achieve a desired overall speed.

[0016] BERT systems typically generate Pseudo Random Bit-Streams (PRBS) to evaluate DUTs. A PRBS may have 2^7, 2^9, 2^{11}, 2^{31} and other bitstreams. BERT systems may also generate Pseudo Random Word Sequences (PRWS). As test patterns, which may be in the form of bitstreams and word sequences, increase in size, the overhead incurred in maintaining deskew or synchronization information for the test pattern increases in the form of processing requirements and memory requirements increases.

[0017] In traditional systems, deskew data for outgoing data is prepared along with the outgoing data such that the preparation of outgoing data and outgoing deskew information are integrated. If a memory-based independent deskew lane is used in which deskew data is stored in and retrieved from a memory, the test pattern size or length is limited by the size of the memory. As set forth herein, deskew data is achieved independently of the preparation of outgoing transmission data and is generated on the fly according to pattern definitions. Similarly, when received data is evaluated, the preparation of deskew information is achieved independently from the processing of received data and on the fly according to pattern definitions.

[0018] The testing system 110 may include software and hardware for providing the functionality and features of the invention. The hardware and firmware components of the

testing system 110 may include various specialized units, circuits, software and interfaces for providing the functionality and features of the invention.

[0019] The testing system 110 may include a back plane 112 to which lane cards 114 and a deskew card 116 are coupled. In one embodiment, there is a single FPGA per data lane and an FPGA for the deskew lane. By allocating an FPGA per lane, independent test patterns may be used per lane. Further, by allocating an FPGA per lane, arbitrarily long test patterns may be used without limitation. The use of independent test patterns per lane and arbitrarily long test patterns provides for improved testing and evaluation of DUTs by allowing for more robust testing of DUTs, such as testing for a greater number of test scenarios and testing with a larger amount of data than is available in traditional systems.

[0020] FPGAs 120 and 122 may be included on the lane cards 114 and the deskew card 116. In one embodiment, there are four lane cards 114 and one deskew card 116. In this embodiment, each lane card 114 includes two lane FPGAs 120, each lane card 114 supports four data lanes 126, and each lane FPGA 120 may support two data lanes 126. Other embodiments may have more or fewer lane FPGAs on more or fewer lane cards, and each lane FPGA may support 1, 2, 3, 4 or more lanes. The more lane FPGAs per lane card, the fewer lane cards are needed to provide the 16 data lanes required by SFI-5. The lane cards 114 each operate in a synchronized interleaved fashion in coordination with the deskew card 116 to transmit data over lines 126.

[0021] The deskew card 116 transmits information on the deskew lane 128 that is used to synchronize and align the information the lane cards 114 transmit on the data lanes 126. The deskew card 116 may comprise a deskew FPGA 122 and a processor. The deskew card 116 may support a single deskew lane 128. The deskew card 116 independently creates deskew

information, also known as synchronization information, without any communication with the lane cards 114 or monitoring of the lane data on the data lanes 126. The deskew card 116 may also include random access memory (RAM), a read-only memory (ROM), programmable read-only memory (PROM), EEPROM or other storage media on which pattern definitions, such as definitions for PRBS patterns, may be stored. In one embodiment, the deskew FPGA 122 includes a RAM block and a ROM block. In another embodiment, the deskew FPGA 122 is coupled to a ROM to access the PRBS pattern definitions or other pattern information. The test pattern definitions may be transferred to the deskew card 116 from a storage medium external to the testing system 110, such as from computer 100, and stored in the deskew FPGA 122 or RAM included on deskew card 116.

[0022] In other implementations, the FPGAs 120 and 122 may be replaced with other silicon devices (*e.g.*, ASICs, PLAs, PLDs, and others). The FPGAs 120 and 122 may also be replaced with software that executes on a microprocessor.

[0023] The computer 100, which may be, for example, a personal computer, may be coupled to the testing system 110. In one embodiment in which the lane cards 114 and deskew card 116 of testing system 110 include FPGAs or other programmable components, the computer 100 transfers software, object code, and/or other instructions, etc. to the FPGAs or other programmable components on the lane cards 114 and the deskew card 116. This transfer may occur regularly on startup of testing system 110, or only when a new version of the software is provided to computer 100. The software or other instructions may be stored on a hard disk included in the computer 100. The software or other instructions may be available over a

network from a local or remote source, such as, for example, a server computer or external disk drive to computer 100 and/or to the testing system 110.

[0024] Software executed in the computer 100 may provide a management user interface to allow a user to access the testing system 110. This access may include monitoring and controlling the testing system 110. The management user interface may be used to start and stop testing, designate system-defined test patterns for each lane, and notify the deskew card 116 which testing pattern is loaded in each lane or lane card 114. The system-defined test patterns may be a PRBS, PRWS, or other test pattern. The software that provides the management user interface may be stored on and executed from a hard disk drive or other storage device included in or coupled to the computer 100.

[0025] The techniques discussed herein may be implemented with any storage media in any storage device included with or otherwise coupled or attached to the computer 100. These storage media include, for example, magnetic media such as hard disks, floppy disks and tape; optical media such as compact disks (CD-ROM and CD-RW) and digital versatile disks (DVD and DVD±RW); solid state memory such as flash memory; and any other storage media. As used herein, a storage device is a device that allows for reading and/or writing to a storage medium. Storage devices include, hard disk drives, DVD drives, flash memory devices, microdrives, minidrives, and others.

[0026] Additional and fewer units, modules or other arrangement of software, hardware and other components may be used to achieve the testing system described herein.

[0027] FIG. 2 is a block diagram of a lane card 200 that may be included in a testing system, such that the lane card 200 may be used as the lane card 114 of the testing system 110 of FIG. 1. The lane card 200 may comprise a backplane connector 212 and two lane FPGAs 220 and 222. Each lane FPGA 220 and 222 may be coupled to two SERDES interfaces. That is, one lane FPGA 220 may be coupled with the SERDES interfaces 232 and 234, and the other lane FPGA 222 may be coupled with SERDES interfaces 236 and 238. It is via the SERDES interfaces 232, 234, 236 and 238 that the lane FPGAs 220 and 222 communicate with a DUT or SUT. Each of the SERDES interfaces 232, 234, 236 and 238 may operate at 2.5 Gbps. Other speeds may be used to achieve a desired overall speed for the lane card 200 or for each of the lane FPGAs 220 and 222.

[0028] A backplane connector 212 may be used to couple lane card 200 within a testing system. The backplane connector 212 may be used for receipt of electrical power and/or for transmission of and receipt of data. The lane card 200 may receive test patterns and other information over the backplane connector 212 from a computer coupled to a testing system in which the lane card is coupled. The lane card 200 may store test patterns in FPGAs 220 and 222 and/or in RAM and ROM (not shown) which may be included on the lane card 200.

[0029] FIG. 3 is a block diagram of an embodiment of a transmitter circuit 300 for implementing an independent deskew lane. The transmitter circuit 300 may be implemented on an FPGA which may be included on a deskew card, such as deskew card 116. The FPGA may have a pre-loaded RAM block which serves as a ROM. The FPGA may also have a RAM block included therein.

[0030] A lane counter 310 is coupled to and provides input to a pattern select unit 320, a pattern seed unit 332, and a current pattern unit 334. The lane counter 310 increments from 1 through 17 or 0 through 16, and designates for which lane the transmitter is providing deskew data, or whether deskew or other data is being transmitted.

[0031] The pattern select 320 is coupled to and provides input to the pattern seed unit 332, the current pattern generation unit 342, and a next pattern generation unit 346. The pattern select unit 320 outputs a signal that specifies or identifies which pattern is to be transmitted on the lane specified by the lane counter 310. The pattern select unit 320 may be a preloaded RAM block in an FPGA that serves as a ROM, holding a system-defined test pattern definition. The pattern definition or pattern definitions stored in the pattern select unit 320 may be received in response to a user command provided via a management interface to a testing system. That is, in one embodiment, a user selects from a system-defined group of test patterns to be used for testing.

[0032] The pattern seed unit 332 may provide a system-defined number of pattern definitions for each of the 16 data lanes. For example, a testing unit may provide a user the ability to test using system-defined number of test patterns which may be provided by a manufacturer and downloaded to the FPGA on startup.

[0033] The current pattern unit 334 provides the current state for the 16 data lanes. The current state may be based on the value output by the lane counter 310 and derived from the next pattern generation unit 346, described below. The current state may provide the current pattern definition for each of the data lanes. That the current pattern unit 334 receives input from the next pattern generation unit 346 provides for forward error correction.

[0034] The pattern seed data unit 332 and the current pattern unit 334 are coupled to and provide information to a mux 338. The mux 338 receives a pattern start signal 336. The pattern start signal 336 may be generated in response to a user command to begin testing. The mux 338 receives a pattern seed from the pattern seed unit 332 and a current pattern from the current pattern unit 334. The mux 338 selects between the pattern seed 332 and the current patter based on the pattern start signal 336. If the testing system is powering on and being used for an initial test, the pattern seed from the pattern seed unit 332 is selected. When the testing unit has already been started, there will be no pattern start signal 336; in one implementation in which the pattern start signal 336 is a Boolean value, the signal is set to false in this circumstance. When the testing unit is not in pattern start mode, the current pattern from the current pattern unit 334 is selected by the mux 338.

[0035] A current pattern generation unit 342 generates a current pattern based on the pattern select signal 320 and input from the mux 338. That is, the current pattern generation unit 342 generates a current pattern based on the pattern seed from the pattern seed unit 332 or the current pattern from the current pattern unit 334. The current pattern generated from the current pattern generation unit 342 is provided to the next pattern generation unit 346. The current pattern generation unit 342 is coupled to and provides input to a mux 350 and a next pattern generation unit 346.

[0036] The next pattern generation unit 346 provides the next pattern to be generated to the current pattern unit 334. The next pattern generation unit 346 generates the pattern that will be used after the current pattern has completed. The amount of data that is transmitted per pattern

for all lanes may be 17*64 bits; that is, the total umber of lanes, both data and deskew, with 64 bits per lanes, for a total of 1,088 bits.

[0037] The transmitter circuit 300 outputs deskew data 354. The deskew data 354 is compiled by the mux 350. The mux 350 receives input from the lane counter 310, and uses the lane information received from the lane counter 310 to select between outputting a header 348 and the output of the current pattern generation unit 334, the current pattern. The mux 350 transmits the header 348 when the lane counter designates the lane that corresponds to the deskew lane, and transmits the current pattern when the lane counter designates a lane that is a data lane.

[0038] The header 348 provides deskew information, and may be, according to the SFI-5 standard, a 64 bit data item. The header 348 or a portion thereof may be stored in an FPGA and provided during start up of a testing system from a computer attached to the testing system in which the transmitter circuit is included.

[0039] FIG. 4 is a block diagram of an embodiment of a receiver circuit 400 for implementing an independent deskew lane. The receiver circuit 400 may be implemented on an FPGA and included on a deskew card such as deskew card 116. The FPGA may designate internal portions for use as RAM and ROM, and/or may be coupled to RAM and ROM external to the FPGA.

[0040] The receiver circuit 400 may be included on the same deskew card as a transmitter circuit, such as the transmitter circuit 300 shown in FIG. 3. The receiver circuit 400 and the transmitter circuit 300 may be included in a single FPGA. The receiver circuit 400 and the

transmitter circuit 300 may be included on separate FPGAs on separate deskew cards. The receiver circuit 400 and the transmitter circuit 300 may also be incorporated on other silicon devices and chips such as ASICs, PLAs, PLDs, and others.

[0041] Received data, RxData 402, is input on a line and provided to a mux 430, a frame synchronization (sync) unit 412, and a compare unit 460.

[0042] The frame sync unit 412 identifies where data is located in the RxData 402. The frame sync unit may locate where data is located based on the size of a header expected to be prepended to the RxData 402. The header which the frame sync unit 412 seeks may be 64 bits and may conform to the SFI-5 standard. The frame sync unit 412 is coupled to and provides information to a pattern state machine 450 and a lane counter 410.

[0043] When the frame sync unit 412 identifies the data location, the frame sync unit 412 enables the pattern state machine 450 and may send the pattern state machine 450 an enable signal and/or information about where the pertinent data is located in the RxData 402.

[0044] When the location of the data in the RxData 402 is found, the frame sync unit 412 resets the lane counter 410 based on the starting point of the data in the RxData 402.

[0045] The lane counter 410 is coupled to and provides a current lane designation to the pattern select unit 420, the current pattern unit 424, and the pattern state machine 450.

[0046] Based on the lane specified by the lane counter 410, (a) the pattern select 420 looks up or otherwise provides a pattern to be used for the specified lane, (b) the current pattern unit 424 provides the data expected to be included in the RxData 402 to the mux 430, and (c) the

pattern state machine 450 selects between performing testing or searching for data in the RxData 402. That is, the mux 430 selects between the RxData 402 and the current pattern from the current pattern unit 424.

[0047] The current pattern unit 424 may be a RAM block within an FPGA. The current pattern may include data required for 16 data lines. The current pattern unit 424 may receive input from the next pattern generation unit 446. That the current pattern unit 424 receives input from the next pattern generation unit 446 may provide for forward error correction.

[0048] The current pattern unit 424 may receive input from the lane counter 410 which designates over which lane the RxData 402 is being received. The lane counter 410 serves to select which lane of the current pattern data should be used.

[0049] The mux 430 passes either the current pattern 424 or the RxData 402 to the anticipated pattern generation unit 442.

[0050] The anticipated pattern generation unit 442 receives a pattern specification from the pattern select unit 420 and a pattern from the mux 430, either the current pattern from the current pattern unit 424 or the RxData 402. The anticipated pattern generation unit 442 generates an anticipated pattern based on the pattern received from the mux 430. The anticipated pattern generation unit 442 is coupled to and provides information to the next pattern generation unit 446 and the compare unit 460.

[0051] The next pattern generation unit 446 generates the next pattern based on input received from the pattern select unit 420 and the anticipated pattern received from the anticipated pattern generation unit 442.

[0052] The compare unit 460 compares the anticipated generated pattern output from the anticipated pattern generation unit 442 with the RxData 402. The compare unit 460 may provide a match/no match data item and the RxData 402 to the pattern state machine 450. If the comparison shows that the current generated pattern does not correspond with the received data, the device under test has failed to perform without error and/or in accordance with the communications standard being used, such as, for example, SFI-5.

[0053] The pattern state machine 450 receives a frame synchronization signal from the frame sync unit 412. The frame sync unit 412 enables the pattern state machine 450 to send information to mux 430. The pattern state machine 450 serves as a select of or control of the mux 430. The pattern state machine 450 may receive the results of the compare unit 460 in the form of match/no match data item and increment a local counter per lane. The pattern state machine 450 may evaluate whether the number of errors, the error density, is greater than a system-defined error threshold. When the number of errors is greater than the error threshold, the pattern state machine 450 transitions to/remains in a pattern acquisition state, and the mux 430 is instructed to select the RxData 402. When the number of errors is less than or equal to the error threshold, the pattern state machine 450 transitions to/remains in a pattern locked state, and the mux 430 is instructed to select the current pattern from the current pattern unit 424. That is, based on the signal or information provided by pattern state machine 450, the mux 430 selects between the current pattern from the current pattern unit 424 and the RxData 402.

A Method

[0054] FIG. 5 is a flow chart of the actions taken when transmitting data using an independent deskew lane described herein. A lane counter may be started, as shown in block

510. The lane counter may increment from, for example, 0 through 16. A pattern may be selected based on the value of the lane counter, as shown in block 516. The pattern may be a PRBS or other test pattern. A pattern seed is prepared based on the selected pattern and the lane counter, as shown in block 520. This preparation may be achieved by reading, loading or otherwise accessing a stored pattern seed. A current pattern is generated based on the pattern seed and the selected pattern, as shown in block 526. The current generated pattern is transmitted, as shown in block 530. The actions taken regarding blocks 510 through 530 are typically achieved the first time data is prepared for transmission.

[0055] The next pattern is generated based on the selected pattern and the current generated pattern, as shown in block 540. The next generated pattern is stored, as shown in block 544. The next generated pattern may be stored, for example, in a RAM or an a portion of an FPGA.

[0056] The lane counter is incremented, as shown in block 550. A pattern is selected based on the value of the lane counter, as shown in block 554. A seed pattern is prepared based on the selected pattern and the lane counter, as shown in block 560. The current pattern is generated based on the selected pattern and the stored next generated pattern, as shown in block 564. The current generated pattern or a header is transmitted, based on the value of the lane counter, as shown in block 570. When the value of the lane counter corresponds to a deskew lane, a header is sent; when the value of the lane counter corresponds to a data lane, the current generated pattern is transmitted.

[0057] A check may be made to determine whether there is more data in the form of test patterns to be transmitted, as shown in block 580. If there is more data to be transmitted, the

flow of actions continues at block 540. If there is no further data to be transmitted, the flow of actions ceases.

[0058] FIG. 6 is a flow chart of the actions taken when receiving data according using an independent deskew lane described herein. The flow of actions shown in FIG. 6 may be implemented on an FPGA or other device or combination of components having a circuit like that described above regarding FIG. 4.

[0059] A lane counter is started, as shown in block 610. Incoming data is received, as shown in block 616. The incoming data is synchronized, as shown in block 620. This synchronization may be achieved by locating where pertinent data begins after a pre-pended header. A test pattern may be selected based on the value of the lane counter, as shown in block 626. An evaluation may be performed to determine whether to use the incoming data or a stored next generated pattern to generate an anticipated pattern, as shown in block 630. An anticipated pattern is generated based on the evaluation and the selected pattern, as shown in block 636. During the first time through this sequence of actions, there is no stored next generated pattern, so the incoming data is used.

[0060] The generated anticipated pattern is compared with the incoming data, as shown in block 640. If the generated anticipated pattern and the incoming data do not match an error condition exists and may be flagged, as shown in block 646. Error conditions may be stored, reported, communicated, logged or handled in a variety of ways. The error condition may be recognized by user interface management software and reported to a user.

[0061] The next pattern is generated based on the selected pattern and the anticipated generated pattern, as shown in block 650. The next generated pattern is stored, as shown in block 660. The lane counter is incremented, as shown in block 670.

[0062] A check may be made to determine whether there is more incoming data, either or both expected and actual, as shown in block 680. If there is more data, the flow of actions continues at block 616. If there is no further data to be transmitted, the flow of actions ceases.

[0063] With regard to FIGS. 5 and 6, additional and fewer steps may be taken, and the steps as shown may be combined or further refined to achieve the methods described herein. In addition, some or all of the steps may be achieved in an order other than that shown and/or concurrently.

[0064] Although exemplary embodiments of the invention have been shown and described, it will be apparent to those having ordinary skill in the art that a number of changes, modifications, or alterations to the invention as described herein may be made, none of which depart from the spirit of the present invention. All such changes, modifications and alterations should therefore be seen as within the scope of the present invention.